

TESTING PLAN DUNGEONS V2

1. Test to check Initial Player arrows are 3.
2. Check number of caves that contain additional arrows
3. Check that arrows can added to both caves and tunnels.
4. Check picking up an arrow increases the arrows the player has.
5. Check picking an arrow removes the arrow from cave.
6. Check shooting an arrow reduces the players arrow count.
7. Check dungeon has enough caves to add the specified number of Otyughs.
8. Check the correct number of Otyughs are being added to the dungeon.
9. Check Otyughs are only added to caves.
10. Check that there is an Otyugh in the end cave.
11. Check that there is no Otyugh at the start cave.
12. Check that a pungent smell is perceived when an Otyugh is 1 cave away.
13. Check a faint smell is perceived when an Otyugh is 2 caves away.
14. Check a pungent smell is perceived when there are multiple Otyughs 2 caves away.
15. Check that there is no smell when there are no Otyughs nearby.
16. Check that negative distance is not given as input.
17. Check that the arrow only travels forward when it is in a cave.
18. Check that the arrow curves when it is in a tunnel.
19. Check that the arrow hits if there is an Otyugh in the destination cave (after arrow travelled the specified distance)
20. Check that arrow misses if there is no Otyugh at the destination cave.
21. Check that the arrow is wasted if it hits a wall before reaching the destination cave.
22. Check that the player cannot shoot if player is out of arrows.
23. Check that hitting an Otyugh twice kills the Otyugh and thus the smell is removed from neighboring caves.
24. Check that the player has a 50% chance of not dying if the caves contains an Otyugh with 50% health.
25. Check that the player dies if player enters a cave with otyugh.
26. Check that the player wins if end cave is reached.
27. Check that no further moves can be made after game ends.
28. Check/validate all inputs passed to the Model.
29. Check that the controller throws an exception if appending to output fails.
30. Check that the controller ends the game if the input stream ends abruptly.
31. Check that the controller appends the correct information to output after any moves are made by the user.
32. Check that the distance is a number before passing it to the model.
33. Check that the directions specified are correct and valid.
34. Check controller calls the appropriate model methods as per required.
35. Check that the controller quits the game when specified.

Note: All tests in the previous model are also included.